CpE 645 Image Processing and Computer Vision

Prof. Hong Man

Department of Electrical and Computer Engineering
Stevens Institute of Technology

STEVENS Institute of Technology

Visual Information Environment Laboratory

Linear Transforms

- 1-D Linear transform:

$$y[k] = \sum_{n=0}^{N-1} x[n] a[k,n], \quad k = [0, N-1]$$

- In matrix form: forward transform $\underline{y} = A\underline{x}$

  inverse transform $\underline{x} = A^{-1}\underline{y}$

- Desired properties of a transform:
  - invertible $A^{-1}\left(A\underline{x}\right) = \underline{x}$
  - energy preserving $\sum_{n=0}^{N-1} |x[n]|^2 = \sum_{k=0}^{N-1} |y[k]|^2$

Linear Transforms

- $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{y}}$ are column vectors, and $\boldsymbol{A}$ is a square matrix $\Rightarrow$ be aware of their indices.

$$\textbf{for } \underline{y} = A\underline{x} \rightarrow \quad \underline{x} = \begin{bmatrix} x[n=0] \\ x[n=1] \\ \vdots \\ x[n=N-1] \end{bmatrix}, \quad \underline{y} = \begin{bmatrix} y[k=0] \\ y[k=1] \\ \vdots \\ y[k=N-1] \end{bmatrix},$$

$$A = \begin{bmatrix} a[k=0,n=0] & a[k=0,n=1] & \cdots & a[k=0,n=N-1] \\ a[k=1,n=0] & a[k=1,k=1] & \cdots & a[k=1,n=N-1] \\ \vdots & \vdots & \ddots & \vdots \\ a[k=N-1,n=0] & a[k=N-1,n=1] & \cdots & a[k=N-1,n=N-1] \end{bmatrix}$$

Linear Transform

- Special matrices
  - $\boldsymbol{A}$ is involutory matrix if $\boldsymbol{A}^{-1}=\boldsymbol{A}$
  - $\boldsymbol{A}$ is orthogonal matrix if $\boldsymbol{A}^{-1}=\boldsymbol{A}^{\mathrm{T}}$
  - $\boldsymbol{A}$ is unitary matrix if $\boldsymbol{A}^{-1}=\boldsymbol{A}^{*\mathrm{T}}$

    where * = conjugate, T = transpose
- For a unitary matrix, we have

$$\underline{y} = A\underline{x} \Rightarrow y[k] = \sum_{n=0}^{N-1} x[n]a[k,n]$$

$$\underline{x} = A^{-1}\underline{y} = A^{*T}\underline{y} \Rightarrow x[k] = \sum_{k=0}^{N-1} y[k]a^{*}[k,n]$$

Linear Transform

- Properties of unitary transform
  - Columns of $\boldsymbol{A}^{*\mathrm{T}}$, i.e. $\underline{\boldsymbol{a}}_k = \{\boldsymbol{a}[\boldsymbol{k},\boldsymbol{n}], \mathbf{0} \le \boldsymbol{n} \le \boldsymbol{N}\textbf{-1}\}^{\mathrm{T}}$, are called the basis vectors of $\boldsymbol{A}$.
  - Energy preservation: input energy = output energy

    $\underline{\boldsymbol{x}}^{*\mathrm{T}} \cdot \underline{\boldsymbol{x}} = (\boldsymbol{A}^{*\mathrm{T}}\,\underline{\boldsymbol{y}})^{*\mathrm{T}} \cdot (\boldsymbol{A}^{*\mathrm{T}}\,\underline{\boldsymbol{y}}) = \underline{\boldsymbol{y}}^{*\mathrm{T}}\,\boldsymbol{A} \cdot \boldsymbol{A}^{*\mathrm{T}}\,\underline{\boldsymbol{y}} = \underline{\boldsymbol{y}}^{*\mathrm{T}} \cdot \underline{\boldsymbol{y}}$

    because $\boldsymbol{A} \cdot \boldsymbol{A}^{*\mathrm{T}} = \boldsymbol{I}$ (identity matrix)
  - Energy compaction: most unitary transforms can compact spatial (input) energy into the first a few elements (coefficients) of the transform, and produce decorrelated transform elements (coefficients).

2-D Linear Transform

- 2-D linear transform:

$$\text{forward transform}: y[k_1, k_2] = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} x[n_1, n_2]\, a[k_1, k_2, n_1, n_2],$$

$$\text{inverse transform}: x[n_1, n_2] = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} y[k_1, k_2]\, \tilde{a}[k_1, k_2, n_1, n_2]$$

where $k_1, k_2, n_1, n_2 \in [0, N-1]$, and $\tilde{a}$ is the elements of A^{-1}.

- If A is unitary:

$$\text{inverse transform}: x[n_1, n_2] = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} y[k_1, k_2]\, a^*[k_1, k_2, n_1, n_2]$$

2-D Linear Transform

- If the transform is separable:

$$a[k_1,k_2,n_1,n_2] = a_1[k_1,n_1] \cdot a_2[k_2,n_2]$$

$$y[k_1,k_2] = \sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1} x[n_1,n_2]a_1[k_1,n_1]a_2[k_2,n_2]$$

$$= \sum_{n_1=0}^{N-1} a_1[k_1,n_1]\left(\underbrace{\sum_{n_2=0}^{N-1} x[n_1,n_2]a_2[k_2,n_2]}_{\text{a 1-D transform over } n_2 \Rightarrow \tilde{x}[n_1,k_2]}\right)$$

$$= \underbrace{\sum_{n_1=0}^{N-1} a_1[k_1,n_1]\,\tilde{x}[n_1,k_2]}_{\text{another 1-D transform over } n_1}$$

2-D Linear Transform

- A separable unitary 2-D transform can be written as:

  forward transform : $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}\boldsymbol{A}^{T},$

  inverse transform : $\boldsymbol{X} = \boldsymbol{A}^{*T}\boldsymbol{Y}\boldsymbol{A}^{*}$

  where $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{A}$ are all with size of $\boldsymbol{N} \times \boldsymbol{N},$

  and $\boldsymbol{A}\boldsymbol{A}^{*T} = \boldsymbol{A}^{T}\boldsymbol{A}^{*} = \boldsymbol{I}.$

- Note that $\boldsymbol{Y}=\boldsymbol{A}\boldsymbol{X}\boldsymbol{A}^{\mathrm{T}} \Rightarrow \boldsymbol{Y}=\boldsymbol{A}[\boldsymbol{A}\boldsymbol{X}^{\mathrm{T}}]^{\mathrm{T}}$, which means that this transform can be implemented by first transforming each column of $\boldsymbol{X}$, and then transforming each row of the resulting matrix from the first stage.

2-D Linear Transform

- Let $\underline{\boldsymbol{a}}_{k_1}^{*T} = \{a^*[k_1, n_1], 0 \le n_1 \le N-1\}$ be the k_1^{th} column in A^{*T}
  Let $\underline{\boldsymbol{a}}_{k_2}^{*T} = \{a^*[k_2, n_2], 0 \le n_2 \le N-1\}$ be the k_2^{th} column in A^{*T}
  the matrices $A^*_{k_1,k_2} = \underline{\boldsymbol{a}}_{k_1}^{*T} \cdot \underline{\boldsymbol{a}}_{k_2}^{*}$ are $N \times N$ matrices and they are called the basis images of this transform.

- There is one basis image for each $[\mathbf{n}_1, \mathbf{n}_2]$ pair. The input image $\boldsymbol{X}$ can be viewed as a weighted sum of all the basis images. The weighting is determined by its transform **Y**.

Discrete Fourier Transform

- Recall DTFT

$$X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n}$$

- Uniformly sample $X(e^{j\omega})$ over ω between

  $0 \le \omega \le 2\pi$ at $\omega_k = 2\pi k/N$ for $0 \le k \le N\text{-}1$

$$X[k] = X(e^{j\omega})\Big|_{\omega=2\pi k/N} = \sum_{k=0}^{N-1} x[n]e^{-j2\pi k/N}, \; 0 \le k \le N-1$$

- The N-point sequence $X[k]$ is called the **Discrete Fourier Transform** (DFT) of the finite duration sequence $x[n]$

Discrete Fourier Transform

$$\text{DFT}: X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}, \quad 0 \le k \le N-1$$

$$\text{Inverse DFT}: x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}, \; 0 \le n \le N-1$$

$$\text{Let } W_N = e^{-j\frac{2\pi}{N}} \text{ (Twiddle factor)}$$

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \qquad x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] W_N^{-kn}$$

DFT Example 1

- Given

$$x[n] = \begin{cases} 1 & n = 0 \\ 0 & 1 \le n \le N-1 \end{cases}$$

- DFT

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} = x[0] W_N^0 = 1$$

$$0 \le k \le N-1$$

DFT Example 2

- Given

$$y[n] = \begin{cases} 1 & n = m \\ 0 & 0 \le n \le m-1,\ m+1 \le n \le N-1 \end{cases}$$

- DFT

$$Y[k] = \sum_{n=0}^{N-1} y[n] W_N^{kn} = y[m] W_N^{km} = W_N^{km}$$

$$0 \le k \le N-1$$

DFT Example 3

- Given $$x[n] = \cos(2\pi rn / N), \quad \text{for } 0 \le n \le N-1$$ and the constant r $\;0 \le r \le N-1$
- We have $$x[n] = \frac{1}{2}\left(e^{j2\pi rn/N} + e^{-j2\pi rn/N}\right)$$ $$= \frac{1}{2}\left(W_N^{-rN} + W_N^{rN}\right)$$
- DFT $$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}$$ $$= \frac{1}{2}\left(\sum_{n=0}^{N-1} W_N^{-(r-k)n} + \sum_{n=0}^{N-1} W_N^{(r+k)n}\right) \quad 0 \le k \le N-1$$

DFT Example 3 (*cont*)

- For $W_N = e^{-j\frac{2\pi}{N}}$ we have

$$\sum_{n=0}^{N-1} W_N^{-(k-l)n} = \begin{cases} N & \text{if } k-l = rN, \text{ for any integer } r \\ 0 & \text{otherwise} \end{cases}$$

- We have the DFT

$$X[k] = \begin{cases} N/2 & k = r \\ N/2 & k = N-r \\ 0 & \text{otherwise in } 0 \le k \le N-1 \end{cases}$$

DFT Example 4

$$X[k]=\sum_{n=0}^{N-1}x[n]e^{-j\frac{2\pi}{N}kn},\quad 0\le k\le N-1$$

If $N=4$, Then we have

$$X[k=0]=x[n=0]e^{-j\frac{2\pi}{4}(k=0)(n=0)}+x[n=1]e^{-j\frac{2\pi}{4}(k=0)(n=1)}+x[n=2]e^{-j\frac{2\pi}{4}(k=0)(n=2)}+x[n=3]e^{-j\frac{2\pi}{4}(k=0)(n=3)}$$

$$X[k=1]=x[n=0]e^{-j\frac{2\pi}{4}(k=1)(n=0)}+x[n=1]e^{-j\frac{2\pi}{4}(k=1)(n=1)}+x[n=2]e^{-j\frac{2\pi}{4}(k=1)(n=2)}+x[n=3]e^{-j\frac{2\pi}{4}(k=1)(n=3)}$$

$$X[k=2]=x[n=0]e^{-j\frac{2\pi}{4}(k=2)(n=0)}+x[n=1]e^{-j\frac{2\pi}{4}(k=2)(n=1)}+x[n=2]e^{-j\frac{2\pi}{4}(k=2)(n=2)}+x[n=3]e^{-j\frac{2\pi}{4}(k=2)(n=3)}$$

$$X[k=3]=x[n=0]e^{-j\frac{2\pi}{4}(k=3)(n=0)}+x[n=1]e^{-j\frac{2\pi}{4}(k=3)(n=1)}+x[n=2]e^{-j\frac{2\pi}{4}(k=3)(n=2)}+x[n=3]e^{-j\frac{2\pi}{4}(k=3)(n=3)}$$

DFT Example 4 (cont)

We know that

$$e^{-j\frac{2\pi}{4}(k=0)(n=0)}=1,\ e^{-j\frac{2\pi}{4}(k=0)(n=1)}=1,\ e^{-j\frac{2\pi}{4}(k=0)(n=2)}=1,\ e^{-j\frac{2\pi}{4}(k=0)(n=3)}=1.$$

$$e^{-j\frac{2\pi}{4}(k=1)(n=0)}=1,\ e^{-j\frac{2\pi}{4}(k=1)(n=1)}=-j,\ e^{-j\frac{2\pi}{4}(k=1)(n=2)}=-1,\ e^{-j\frac{2\pi}{4}(k=1)(n=3)}=j.$$

$$e^{-j\frac{2\pi}{4}(k=2)(n=0)}=1,\ e^{-j\frac{2\pi}{4}(k=2)(n=1)}=-1,\ e^{-j\frac{2\pi}{4}(k=2)(n=2)}=1,\ e^{-j\frac{2\pi}{4}(k=2)(n=3)}=-1.$$

$$e^{-j\frac{2\pi}{4}(k=3)(n=0)}=1,\ e^{-j\frac{2\pi}{4}(k=3)(n=1)}=j,\ e^{-j\frac{2\pi}{4}(k=3)(n=2)}=-1,\ e^{-j\frac{2\pi}{4}(k=3)(n=3)}=-j.$$

Hint: always remember:

$$e^{j\omega t}=\cos\omega t+j\sin\omega t$$

DFT Example 4 (cont)

Assume input signal $x[n] = \{1, -1, 3, -2\}$

i.e. $x[0] = 1,\ x[1] = -1,\ x[2] = 3,\ x[3] = -2$

Then we have

$$X[0] = 1\times 1 + (-1)\times 1 + 3\times 1 + (-2)\times 1 = 1$$

$$X[1] = 1\times 1 + (-1)\times(-j) + 3\times(-1) + (-2)\times j = -2 - j$$

$$X[2] = 1\times 1 + (-1)\times(-1) + 3\times 1 + (-2)\times(-1) = 7$$

$$X[3] = 1\times 1 + (-1)\times j + 3\times(-1) + (-2)\times(-j) = -2 + j$$

Circular Shift

- In DFT, a N-point sequence $\boldsymbol{x[n]}$ is defined in $\mathbf{0 \le n \le N\text{-}1}$, and all samples outside this range are considered to be zeros.
- If $\boldsymbol{x[n]}$ is shifted in time, i.e. $\boldsymbol{x_1[n]=x[n-n_0]}$, the defined range is no longer $\mathbf{0 \le n \le N\text{-}1}$, and therefore we can not take direct DFT of $\boldsymbol{x_1[n]}$.
- **Circular shift** is defined as a shift which will produce a new sequence that is still in the same range $\mathbf{0 \le n \le N\text{-}1}$.
- The traditional shift is then called **linear shift**.

Circular Shift

- Circular shift is denoted as $x_1[n]=x[((n-m))_N]$.
- For $m \geq 0$ – right shift

$$x_1[n] = \begin{cases} x[n+N-m], & \text{for } 0 \leq n < m \\ x[n-m], & \text{for } m \leq n \leq N-1 \end{cases}$$

- For $m<0$ – left shift

$$x_1[n] = \begin{cases} x[n-m], & \text{for } 0 \leq n < N-m \\ x[n-(N-m)], & \text{for } N-m \leq n < N-1 \end{cases}$$

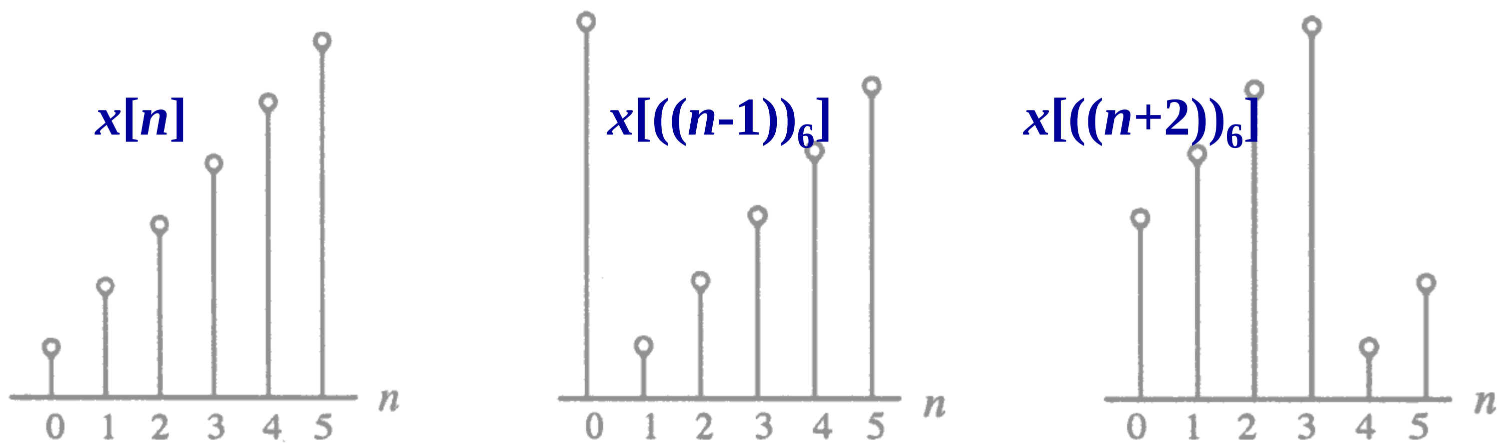
Circular Shift

Therefore, if the N samples of $x[n]$ are

$$(x[0]\ x[1]\ x[2]\ x[3]\ \ldots\ x[N-3]\ x[N-2]\ x[N-1]),$$

the N samples of $x_1[n]$ for $m=2$ are

$$(x[N-2]\ x[N-1]\ x[0]\ x[1]\ x[2]\ x[3]\ \ldots\ x[N-3]).$$



Circular Shift Property

- For $x_1[n]=x[((n-m))_N]$,
  - Right shift by m is equivalent to left shift by $(N-m)$
  - Circular shift by $m+rN$ is equivalent to circular shift by m.

DFT Properties

Linearity : $a\, x_1[n] + b\, x_2[n] \xleftrightarrow{DFT} a\, X_1[k] + b\, X_2[k]$

for $N \geq \max(N_1, N_2)$

Time shift : $x[((n-m))_N] \xleftrightarrow{DFT} W_N^{km}\, X[k]$

Frequency shift : $W_N^{-ln}\, x[n] \xleftrightarrow{DFT} X[((k-l))_N]$

Circular convolution :

$$x_3[n] = x_1[n] *_N x_2[n] = \sum_{r=0}^{N-1} x_1[r]\, x_2[((n-r))_N] \xleftrightarrow{DFT} X_1[k] X_2[k]$$

Modulation : $x_1[n] x_2[n] \xleftrightarrow{DFT} \frac{1}{N} \sum_{r=0}^{N-1} X_1[r]\, X_2[((k-r))_N]$

Duality : $X[n] \xleftrightarrow{DFT} N x[((-k))_N]$

DFT Properties

- Assume $x^*[n]$ is the complex conjugate of $x[n]$

$$x^*[n] \xleftrightarrow{\text{DFT}} X^*[(-k)_N]$$

$$x^*[(-n)_N] \xleftrightarrow{\text{DFT}} X^*[k]$$

- If $x[n]$ is real

$$x[n] = x^*[n] \xleftrightarrow{\text{DFT}} X[k] = X^*[(-k)_N]$$

This is the symmetric property of DFT

DFT and Linear Convolution

Convolution by DFT: (L-point $x_1[n]$; P-point $x_2[n]$)

1. Choose smallest value of N, **N=L+P-1**
2. Compute N-DFT of $x_1[n]$ ➔ $X_1[k]$
3. Compute N-DFT of $x_2[n]$ ➔ $X_2[k]$
4. Compute $X_3[k] = X_1[k]\, X_2[k]$ ➔ $X_3[k]$
5. Compute Inverse N-DFT of $X_3[k]$ ➔ $x_3[n]$

   where $x_3[n]$ should be equal to $x_1[n] * x_2[n]$

2-D Discrete Fourier Transform

- The N×N 2-D DFT is defined as

$$Y[k_1,k_2]=\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1}x[n_1,n_2]e^{-j\frac{2\pi}{N}n_1k_1}e^{-j\frac{2\pi}{N}n_2k_2}$$

and the inverse DFT is defined as

$$x[n_1,n_2]=\frac{1}{N^2}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}Y[k_1,k_2]e^{j\frac{2\pi}{N}n_1k_1}e^{j\frac{2\pi}{N}n_2k_2}$$

for $0\le k_1,k_2,n_1,n_2\le N-1$

2-D DFT Properties

- 2-D DFT is separable:
  - A 2-D DFT can be obtained through
    - first apply 1-D DFT to all the rows (or columns),
    - second apply 1-D DFT to all the resulting columns (or rows).
- Periodic extension in both space and frequency
  - $\mathbf{x[n_1+N,n_2+N]=x[n_1,n_2]}$
  - $\mathbf{Y[k_1+N,k_2+N]=Y[k_1,k_2]}$

2-D DFT Properties

- DFT can be viewed as sampled DTFT
  - If $\tilde{x}[n_1, n_2] = \begin{cases} x[n_1, n_2] & \text{for } 0 \le n_1, n_2 \le N-1 \\ 0 & \text{otherwise} \end{cases}$

$$x[n_1, n_2] \xleftrightarrow{\text{DFT}} Y[k_1, k_2] \text{ and } \tilde{x}[n_1, n_2] \xleftrightarrow{\text{DTFT}} \tilde{Y}(\omega_1, \omega_2)$$

We see that $Y[k_1, k_2] = \tilde{Y}(\omega_1, \omega_2)\,\big|_{(\omega_1 = 2\pi k_1/N,\ \omega_2 = 2\pi k_2/N)}$

  - Therefore

k_1, k_2	ω_1, ω_2	
0	0	zero frequency (DC)
N/2	π	highest frequency
N-1	$(N-1/N)2\pi \approx 2\pi$	low frequency

2-D DFT Properties

- Conjugate Symmetric

  For a real $\boldsymbol{x}[\boldsymbol{n}_1, \boldsymbol{n}_2]$

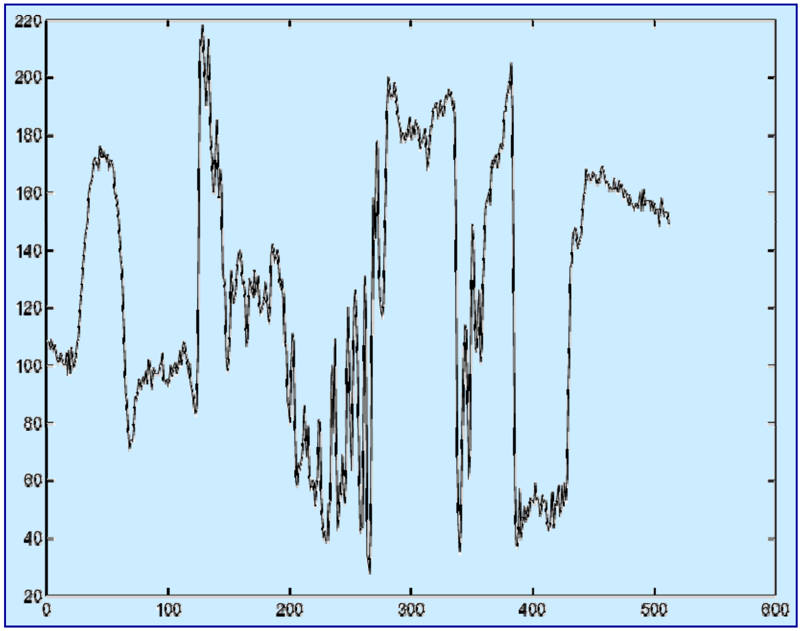
  $$\boldsymbol{Y}\left[\frac{\boldsymbol{N}}{\mathbf{2}} \pm \boldsymbol{k}_1, \frac{\boldsymbol{N}}{\mathbf{2}} \pm \boldsymbol{k}_2\right] = \boldsymbol{Y}^*\left[\frac{\boldsymbol{N}}{\mathbf{2}} \mp \boldsymbol{k}_1, \frac{\boldsymbol{N}}{\mathbf{2}} \mp \boldsymbol{k}_2\right] \text{ for } \mathbf{0} \le \boldsymbol{k}_1, \boldsymbol{k}_2 \le \frac{\boldsymbol{N}}{\mathbf{2}} - \mathbf{1}$$

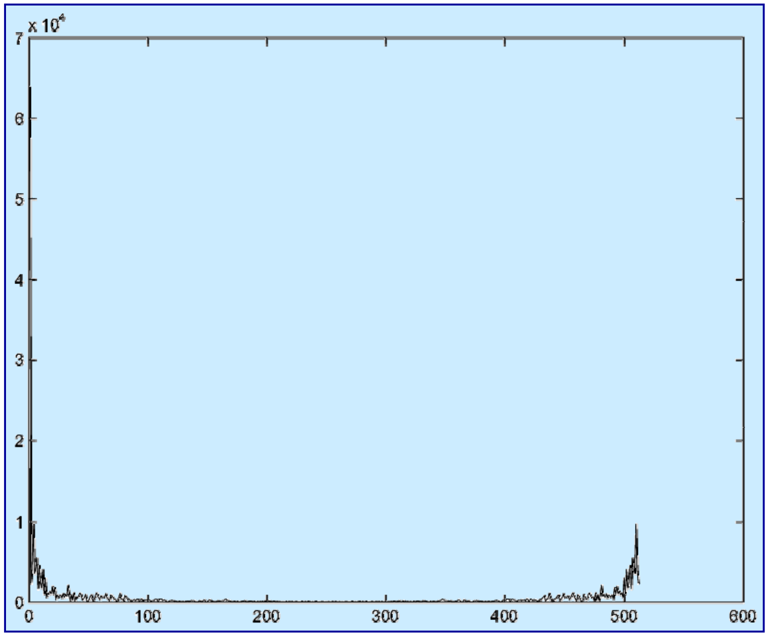
  or

  $$\boldsymbol{Y}[\boldsymbol{k}_1, \boldsymbol{k}_2] = \boldsymbol{Y}^*[N - \boldsymbol{k}_1, N - \boldsymbol{k}_2] \quad \text{for} \quad \mathbf{0} \le \boldsymbol{k}_1, \boldsymbol{k}_2 \le N - \mathbf{1}$$

  – Note that the DC value when $\mathbf{k_1}$, $\mathbf{k_2} = \mathbf{0}$ does not have an associated symmetric point.

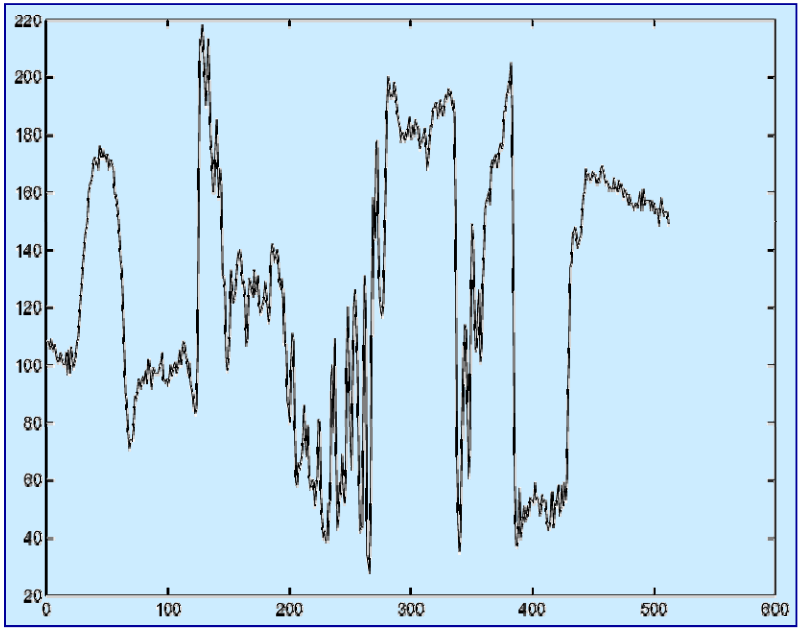
1-D DFT Transform: Example

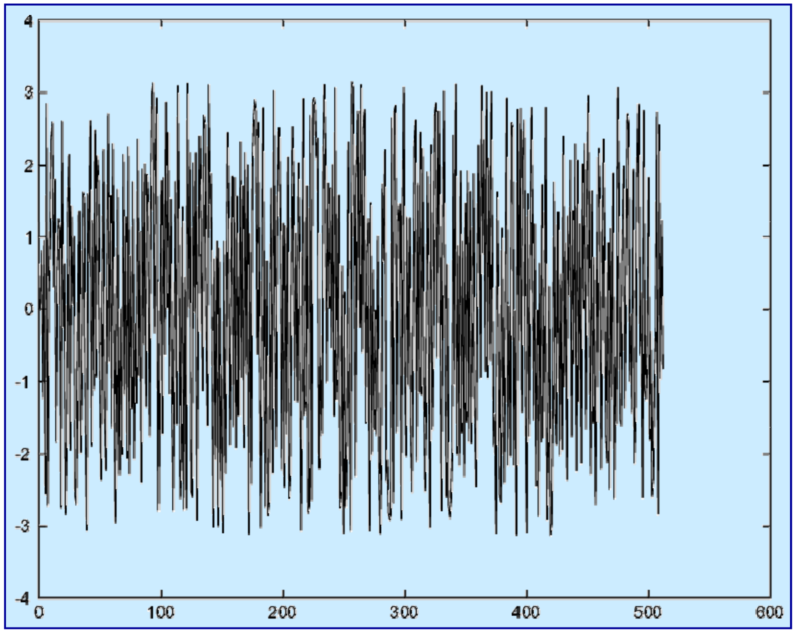




A spatial domain signal and the magnitude of its DFT transformed coefficients.

1-D DFT Transform: Example (*cont.*)

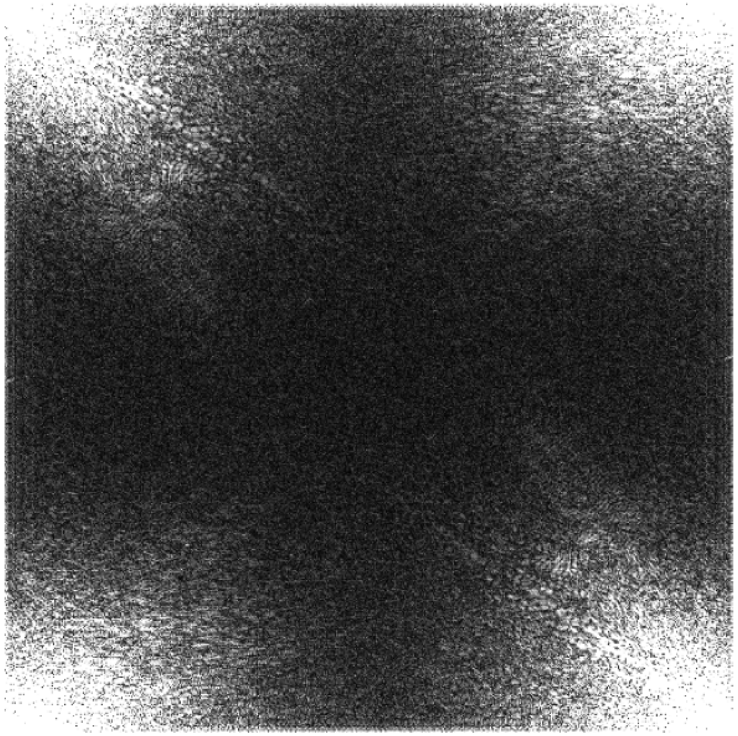




A spatial domain signal and the phase of its DFT transformed coefficients.

2-D DFT Transform: Example

Original Image



2-D DFT transform Magnitude

The Discrete Cosine Transform

- The 1-D discrete cosine transform (DCT) is defined as:

$$y[k] = \alpha[k] \sum_{n=0}^{N-1} x[n] \cos\left[\frac{\pi(2n+1)k}{2N}\right] \quad 0 \le k \le N-1$$

$$\text{where} \quad \alpha[0] = \sqrt{\frac{1}{N}} \qquad \alpha[k] = \sqrt{\frac{2}{N}} \quad 1 \le k \le N-1$$

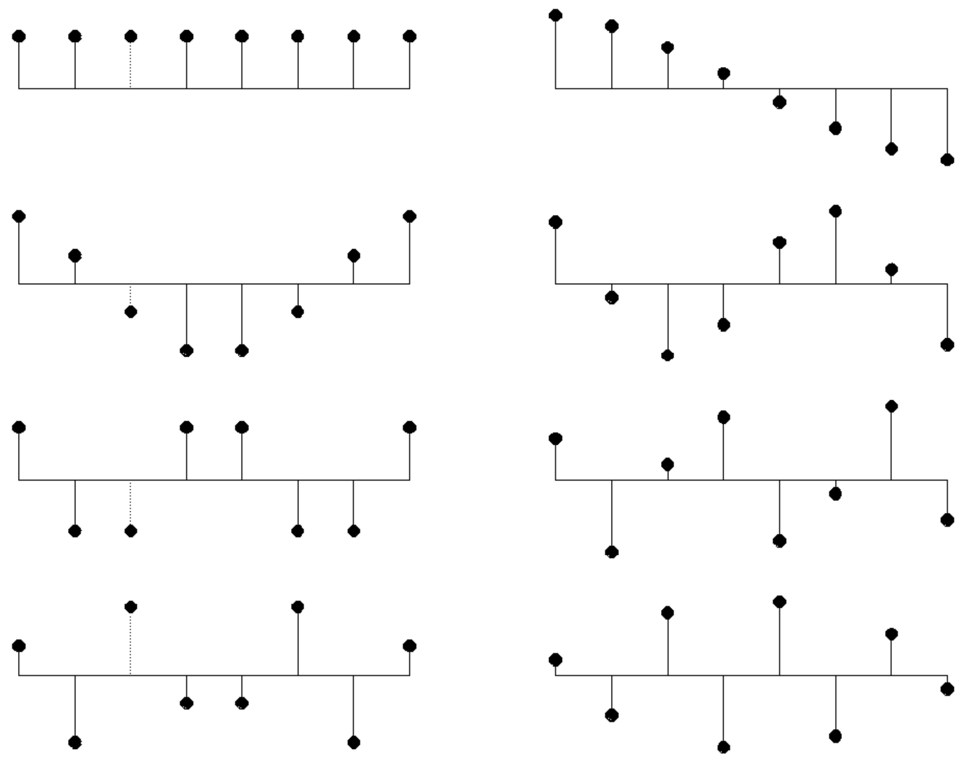
- DCT belongs to the family of sinusoidal transforms. Its basis functions are sinusoidal waveforms.
- It can be computed from the DFT. Therefore it has fast algorithm derived from the FFT.

The DCT Properties

- Properties of the DCT:
  - It is data independent.
  - It has a near optimal data decorrelation capability. Its performance is especially good for first order Markov models with correlation coefficient ***p*>0.9**.
  - The DCT transform coefficients and the basis functions have frequency-domain interpretations similar to the DFT. Therefore each DCT coefficient represent a certain frequency component in the original signal.
  - Fast transform algorithm exists, $O(N\log_2 N)$.
  - Used in most image/video coding standards, e.g. JPEG, MPEG, Motion-JPEG, etc.

The DCT Basis Vectors (N=8)

- The basis functions of the DCT are real sinusoids.



2-D Discrete Cosine Transform

- The N×N 2-D DCT is defined as

$$Y[k_1,k_2]=\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1}c_1(k_1)c_2(k_2)x[n_1,n_2]\cos\left[\frac{(2n_1+1)k_1\pi}{2N}\right]\cos\left[\frac{(2n_2+1)k_2\pi}{2N}\right],$$

and the inverse 2-D DCT is defined as

$$x[n_1,n_2]=\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}c_1(k_1)c_2(k_2)y[k_1,k_2]\cos\left[\frac{(2n_1+1)k_1\pi}{2N}\right]\cos\left[\frac{(2n_2+1)k_2\pi}{2N}\right]$$

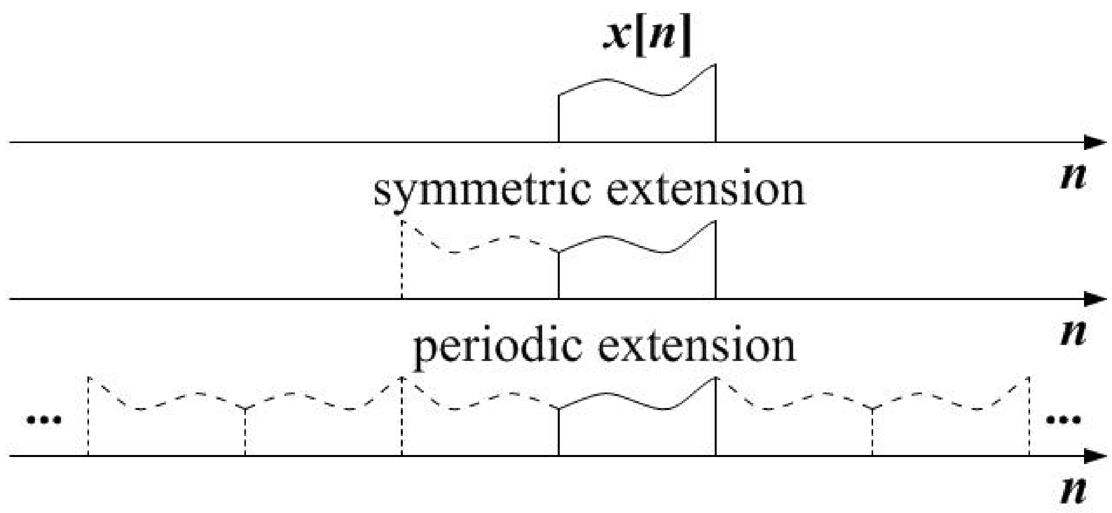
where $c_1(k_1)=\begin{cases}\frac{1}{\sqrt{N}} & \text{for } k_1=0\\ \sqrt{\frac{2}{N}} & \text{otherwise}\end{cases}$ and $c_2(k_2)=\begin{cases}\frac{1}{\sqrt{N}} & \text{for } k_2=0\\ \sqrt{\frac{2}{N}} & \text{otherwise}\end{cases}$

2-D DCT Properties

- DCT is real in both spatial and transformed domain.
- 2-D DCT is Separable
  - A 2-D DCT can be obtained through
    - first applying 1-D DCT to each of the rows (or columns),
    - second applying 1-D DCT to each of the resulting columns (or rows).

2-D DCT Properties

- DCT is related to DFT
  - Consider the 1-D case, given discrete signal ***x*[*n*],** we can a new signal $\tilde{x}[n] = (x[n] + x[-n-1])_{2N}$ which is a symmetric and periodic extension of ***x*[*n*]** with period of **2N.**



2-D DCT Properties

– Then take DFT of $\tilde{x}[n]$, we have

$$\tilde{y}[k] = \sum_{n=0}^{2N-1} \left(\tilde{x}[n] e^{-j\frac{2\pi}{2N}nk} \right), \text{ for } 0 \le n \le 2N-1 \text{ and } 0 \le k \le 2N-1$$

– If we modulate the $\tilde{y}[k]$ with $e^{-j\frac{2\pi}{2N}\cdot\frac{1}{2}k}$

$$e^{-j\frac{2\pi}{2N}\cdot\frac{1}{2}k} \cdot \tilde{y}[k]$$

$$= \sum_{n=0}^{2N-1} \left(\left(x[n] + x[-n-1] \right)_{2N} e^{-j\frac{2\pi}{2N}(n+\frac{1}{2})k} \right)$$

$$= \sum_{n=0}^{2N-1} \left(\underbrace{\left(x[n] + x[-n-1] \right)_{2N}}_{\substack{\text{even around} \\ n = -½}} \cdot \left(\underbrace{\cos \frac{2\pi}{2N}(n+\frac{1}{2})k}_{\substack{\text{even around} \\ n = -½}} - \underbrace{j \sin \frac{2\pi}{2N}(n+\frac{1}{2})k}_{\substack{\text{odd around} \\ n = -½}} \right) \right)$$

2-D DCT Properties

– We know that in general

Σ_{∞} even terms × even terms ⇒ survives

Σ_{∞} even terms × odd terms ⇒ 0

– Therefore

$$e^{-j\frac{2\pi}{2N}\cdot\frac{1}{2}k}\cdot\tilde{y}[k]$$

$$=\sum_{n=0}^{2N-1}\left(\left(x[n]+x[-n-1]\right)_{2N}\cdot\left(\cos\frac{2\pi}{2N}(n+\frac{1}{2})k\right)\right)$$

$$=2\cdot\sum_{n=0}^{N-1}\left(x[n]\cdot\cos\frac{2\pi}{2N}(n+\frac{1}{2})k\right)$$

This is almost the 1-D DCT of $\boldsymbol{x[n]}$ for $\mathbf{0 \le n \le N\text{-}1}$

2-D DCT Properties

- Conclusion: the 1-D DCT of of $\boldsymbol{x[n]}$ for $\mathbf{0 \le \boldsymbol{n} \le N\text{-}1}$ can be obtained through

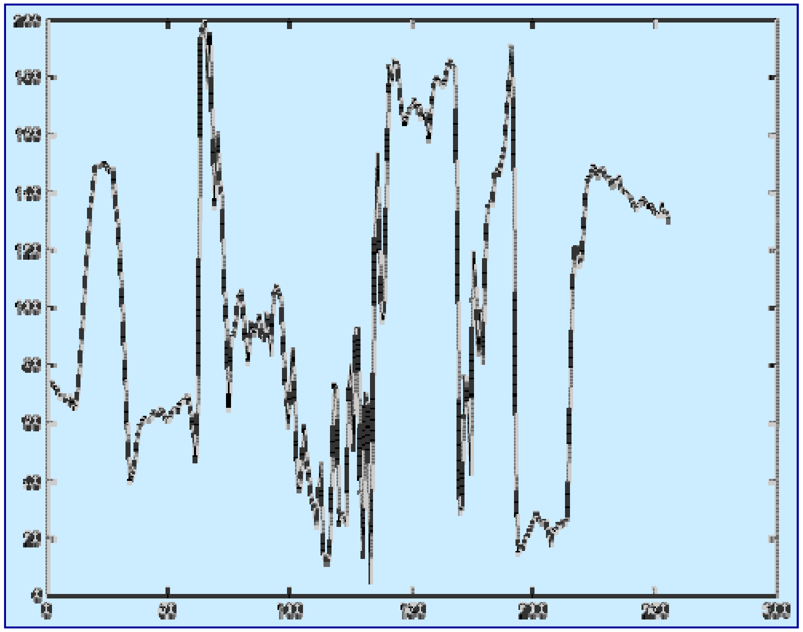
$$e^{-j\frac{\pi}{2N}k} \cdot DFT\left\{ (x[n] + x[-n-1])_{2N} \right\}$$

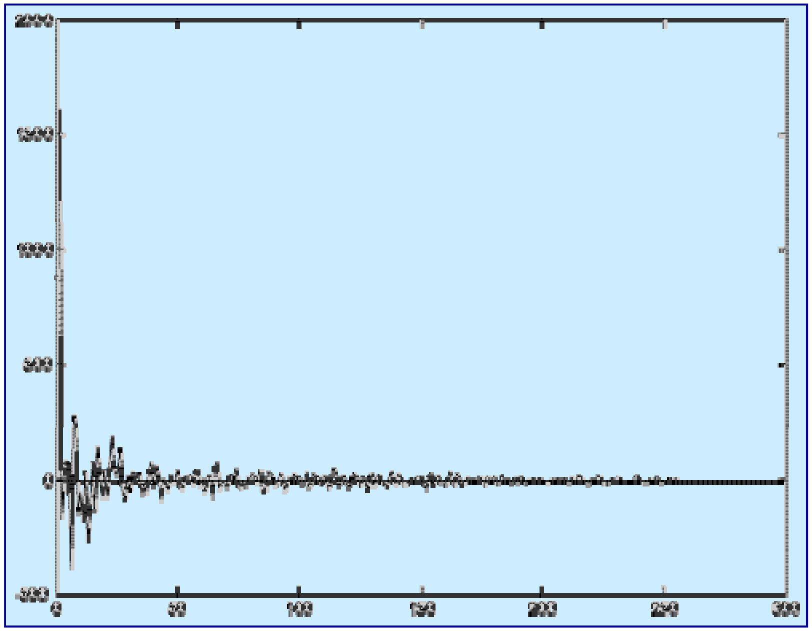
and take only $\mathbf{0 \le \boldsymbol{k} \le N\text{-}1}$

- 2-D DCT is related to 2-D DFT in the sense that it only represent one quadrant of the DFT frequency domain.

$\mathbf{k_1, k_2}$	$\boldsymbol{\omega_1, \omega_2}$	
0	**0**	zero frequency (DC)
N-1	**(N-1/N)π≈π**	highest frequency

1-D DCT Transform: Example

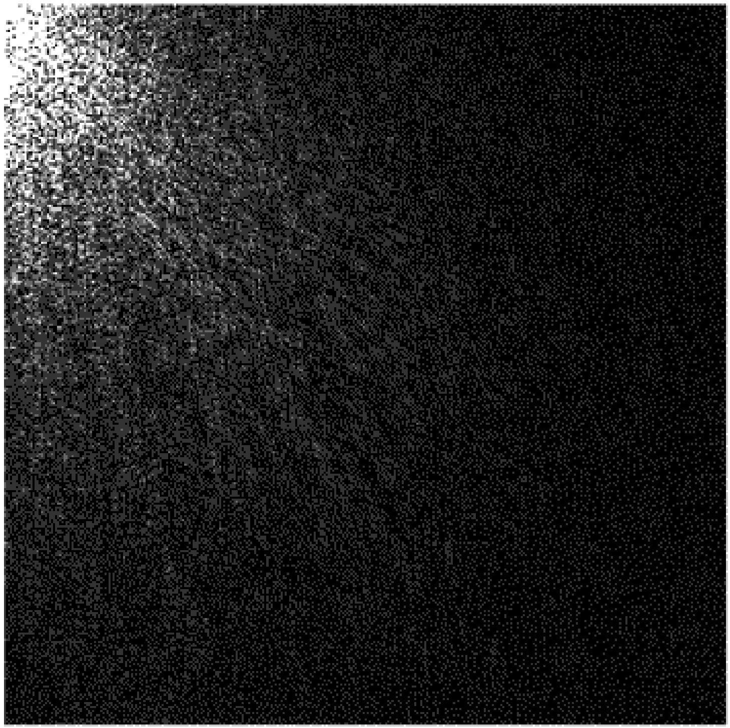




A spatial domain signal and its DCT transformed coefficients.

2-D DCT Transform: Example

Original Image



2-D DCT transform

Karhunen-Loeve Transform

- Most of the popular transforms have the property of energy concentration -- compaction
  - Energy of the transform coefficients is concentrated in a small region, usually around the origin.
- Such a property leads to efficient data representation, which is essential in data analysis and data compression applications. For example:
  - Dimension reduction in feature generation
  - Data de-correlation and redundancy removal in compression
- Here we are seeking the optimal data transform in terms of compaction.

Karhunen-Loeve Transform

- Problem formulation:
  - Assume an orthogonal matrix $\boldsymbol{A}$, i.e. $\boldsymbol{A}^{-1}=\boldsymbol{A}^{T}$
    - The forward transform is defined as $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}$
    - The inverse transform is defined as $\boldsymbol{x}=\boldsymbol{A}^{T}\boldsymbol{y}$
  - Express $\boldsymbol{A}^{T}$ in terms of basis vectors:
    - $\boldsymbol{A}^{T}=[\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{N-1}]$
    - Then each transform coefficient can be expressed as

$$y[k] = \boldsymbol{v}_k^T \boldsymbol{x}, \quad \text{for } k = 0,1,\ldots, N-1$$

    - And $\boldsymbol{x}$ can be expressed as a weighted sum of the basis vectors

$$\boldsymbol{x} = \sum_{k=0}^{N-1} y[k]\boldsymbol{v}_k$$

Karhunen-Loeve Transform

- A measure of energy compaction is to be defined
  - Given an integer M and $M<N$,
  - we consider to use the first M transform coefficients to approximate the original $\boldsymbol{x}$, i.e.

$$\tilde{\boldsymbol{x}} = \sum_{k=0}^{M-1} y[k]\boldsymbol{v}_{\boldsymbol{k}}$$

  - The error of this approximation is

$$\varepsilon = \boldsymbol{x} - \tilde{\boldsymbol{x}} = \sum_{k=0}^{N-1} y[k]\boldsymbol{v}_{\boldsymbol{k}} - \sum_{k=0}^{M-1} y[k]\boldsymbol{v}_{\boldsymbol{k}} = \sum_{k=M}^{N-1} y[k]\boldsymbol{v}_{\boldsymbol{k}}$$

  - We seek transform $\boldsymbol{A}$ that can minimize the average energy of the error term for any given M.

Karhunen-Loeve Transform

- The average energy of the error term can be expressed as

$$E\{\varepsilon^{\mathrm{T}}\varepsilon\} = \boldsymbol{E}\left\{\left(\sum_{k=M}^{N-1} y[k]\mathbf{v}_k\right)^T \left(\sum_{m=M}^{N-1} y[k]\mathbf{v}_m\right)\right\}$$

$$= \sum_{k=M}^{N-1}\sum_{m=M}^{N-1} \mathbf{v}_k^T E\{y[k]y[m]\}\mathbf{v}_m$$

$$= \sum_{k=M}^{N-1}\sum_{m=M}^{N-1} E\{y[k]y[m]\}\mathbf{v}_k^T\mathbf{v}_m$$

$$= \sum_{k=M}^{N-1}\sum_{m=M}^{N-1} E\{y[k]y[m]\}\delta[k-m] \longleftarrow$$

$$= \sum_{k=M}^{N-1} E\{y[k]^2\}$$

for orthogonal transform

$\mathbf{v}_k^T\mathbf{v}_m = \delta[k-m],$

for $\quad k, m = 0, 1, \ldots, N-1$

Karhunen-Loeve Transform

- The average error energy can be further expressed as

$$E\{\varepsilon^T \varepsilon\} = \sum_{k=M}^{N-1} \boldsymbol{E}\left\{(\boldsymbol{v}_k^T \boldsymbol{x})(\boldsymbol{v}_k^T \boldsymbol{x})^T\right\}$$

$$= \sum_{k=M}^{N-1} \boldsymbol{E}\left\{\boldsymbol{v}_k^T \boldsymbol{x}\boldsymbol{x}^T \boldsymbol{v}_k\right\}$$

$$= \sum_{k=M}^{N-1} \boldsymbol{v}_k^T \boldsymbol{E}\left\{\boldsymbol{x}\boldsymbol{x}^T\right\} \boldsymbol{v}_k$$

$$= \sum_{k=M}^{N-1} \boldsymbol{v}_k^T \boldsymbol{R}_{xx} \boldsymbol{v}_k$$

$\boldsymbol{E}\left\{\boldsymbol{x}\boldsymbol{x}^T\right\}$ is the autocorrelation matrix of $\boldsymbol{x}$, denoted as $\boldsymbol{R}_{xx}$, which is real, symmetric and assumed to be positive definite.

Karhunen-Loeve Transform

- We impose an additional constraint to seek an orthonormal transform $\boldsymbol{A}$, i.e.

$$\boldsymbol{v}_k^T \boldsymbol{v}_k = 1$$

- The problem of minimizing $\boldsymbol{E}\{\boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon}\}$ under the constraint of orthonormal condition can be solved by using Lagrange multiplier
  - The Lagrangian

$$J = \sum_{k=M}^{N-1} \boldsymbol{v}_k^T \boldsymbol{R}_{xx} \boldsymbol{v}_k - \lambda_k (\boldsymbol{v}_k^T \boldsymbol{v}_k - 1)$$

  - The solution $\boldsymbol{v}_k$ that minimizes $\boldsymbol{J}$ can be found by

$$\frac{\partial J}{\partial \boldsymbol{v}_k} = 2\boldsymbol{R}_{xx} \boldsymbol{v}_k - 2\lambda_k \boldsymbol{v}_k = 0$$

$$\boldsymbol{R}_{xx} \boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k$$

Karhunen-Loeve Transform

- To find the $\boldsymbol{v_k}$ and λ_k becomes an eigen-decomposition problem,
  - $\boldsymbol{v_k}$'s that satisfy $\boldsymbol{R}_{xx}\boldsymbol{v_k} = \lambda_k \boldsymbol{v_k}$ are the eigenvectors of $\boldsymbol{R_{xx}}$,
  - the λ_k's are the eigenvalues of associated eigenvectors.
- Inserting solution $\boldsymbol{v_k}$'s into the average error energy formula, we have

$$E\{\varepsilon^T \varepsilon\} = \sum_{k=M}^{N-1} \boldsymbol{v}_k^T \boldsymbol{R}_{xx} \boldsymbol{v}_k = \sum_{k=M}^{N-1} \boldsymbol{v}_k^T \boldsymbol{v}_k \lambda_k = \sum_{k=M}^{N-1} \lambda_k$$

$$\boxed{\text{given the condition } \boldsymbol{v}_k^T \boldsymbol{v}_k = 1}$$

Karhunen-Loeve Transform

- This implicates that the eigenvalue λ_k's determine the approximation error energy, which suggests:
  - Sort λ_k's in decreasing order, which also determines the order of the associated $\boldsymbol{v_k}$'s,
  - For any given M, one should use the first M $\boldsymbol{v_k}$'s in the approximation

$$\tilde{\boldsymbol{x}} = \sum_{k=0}^{M-1} y[k]\boldsymbol{v_k}$$

  - This will keep the approximation error energy minimum because

$$E\{\varepsilon^T \varepsilon\} = \sum_{k=M}^{N-1} \lambda_{\boldsymbol{k}}$$

Karhunen-Loeve Transform

- The Karhunen-Loeve Transform (KLT) achieves the optimal performance in energy compaction, because the signal energy is maximally packed to the first M coefficients, and it leads to minimum energy loss if the last N-M coefficients are omitted.
- However KLT requires to estimate the autocorrelation matrix $\boldsymbol{R}_{xx}$ of the input vector $\boldsymbol{x}$, which makes KLT data dependent, i.e. the KLT transform matrix $\boldsymbol{A}$ is different for every different input. This is not desirable in practice.
- On natural audio and image/video signals, DCT is considered to be good substitute of KLT without data dependency.